

## 1. Creating a Dockerfile and building an image from it

```
seansom@alm MINGW64  
$ touch Dockerfile
```

```
1 FROM ubuntu:16.04  
2 LABEL author="Sean Som"  
3  
4  
5 RUN apt-get update  
6 RUN apt-get install -y iputils-ping
```

```
seansom@alm MINGW64 ~/OneDrive/Documents/Programming/Repositories/CoE-197-S/ME8_Containerizat  
$ docker build -t 'seansom/ping' .  
[+] Building 103.7s (7/7) FINISHED  
=> [internal] load build definition from Dockerfile  
=> => transferring dockerfile: 432B  
=> [internal] load .dockerignore  
=> => transferring context: 2B  
=> [internal] load metadata for docker.io/library/ubuntu:16.04  
=> [1/3] FROM docker.io/library/ubuntu:16.04  
=> [2/3] RUN apt-get update  
=> [3/3] RUN apt-get install -y iputils-ping  
=> exporting to image  
=> => exporting layers  
=> => writing image sha256:644dc353fa8babe09473cf9db0a77866bd8e1c4f8e5909d973eb046f8a5368e3  
=> => naming to docker.io/seansom/ping
```

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to f

```
seansom@alm MINGW64 ~/OneDrive/Documents/Programming/Repositories/CoE-197-S/ME8_Containerizat  
$ docker images  
REPOSITORY      TAG         IMAGE ID      CREATED        SIZE  
seansom/ping    latest     644dc353fa8b  43 seconds ago 169MB  
hello-docker    latest     5a8b3718878c  6 days ago    113MB  
ubuntu          16.04      aefd7f02ae24  2 weeks ago   134MB  
ubuntu          latest     7e0aa2d69a15  2 weeks ago   72.7MB
```

## 2. Seeing cache behavior in rebuilding image

```
seansom@alm MINGW64 ~/OneDrive/Documents/Programming/Repositories/CoE-197-S/ME8_Containerization_and_D
$ docker build -t 'seansom/ping' .
[+] Building 0.2s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 432B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/ubuntu:16.04
=> [1/3] FROM docker.io/library/ubuntu:16.04
=> CACHED [2/3] RUN apt-get update
=> CACHED [3/3] RUN apt-get install -y iputils-ping
=> exporting to image
=> => exporting layers
=> => writing image sha256:f022b642f35cca1fd4887d4a58e4b34792e8b20248f366e85983c81f619bd7a2
=> => naming to docker.io/seansom/ping
```

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

### 3. Optimizing RUN commands in the Dockerfile

```
1 FROM ubuntu:16.04
2 LABEL author="Sean Som"
3
4 RUN apt-get update \
5     && apt-get install -y iputils-ping \
6     && apt-get clean \
7     && cd /var/lib/apt/lists && rm -fr *Release* *Sources* *Packages* \
8     && truncate -s 0 /var/log/*log
```

```
seansom@alm MINGW64 ~/OneDrive/Documents/Programming/Repositories
$ docker images
```

| REPOSITORY   | TAG    | IMAGE ID     | CREATED       | SIZE   |
|--------------|--------|--------------|---------------|--------|
| seansom/ping | latest | b036fcb9ca29 | 9 seconds ago | 138MB  |
| hello-docker | latest | 5a8b3718878c | 6 days ago    | 113MB  |
| ubuntu       | 16.04  | aefd7f02ae24 | 2 weeks ago   | 134MB  |
| ubuntu       | latest | 7e0aa2d69a15 | 2 weeks ago   | 72.7MB |

### 4. Adding ENV and CMD directives to Dockerfile and running rebuilt image

```
1 FROM ubuntu:16.04
2 LABEL author="Sean Som"
3
4 ENV PING_TARGET "google.com"
5
6 RUN apt-get update \
7     && apt-get install -y iputils-ping \
8     && apt-get clean \
9     && cd /var/lib/apt/lists && rm -fr *Release* *Sources* *Packages* \
10    && truncate -s 0 /var/log/*log
11
12 CMD ["sh", "-c", "ping $PING_TARGET"]
```

```
seansom@alm MINGW64 ~/OneDrive/Documents/Programming/Repositories
$ docker run -it --rm seansom/ping
PING google.com (216.58.200.78) 56(84) bytes of data.
64 bytes from 216.58.200.78: icmp_seq=1 ttl=37 time=22.4 ms
64 bytes from 216.58.200.78: icmp_seq=2 ttl=37 time=22.2 ms
64 bytes from 216.58.200.78: icmp_seq=3 ttl=37 time=26.9 ms
64 bytes from 216.58.200.78: icmp_seq=4 ttl=37 time=22.7 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 22.201/23.615/26.976/1.960 ms
```